Hierarchical Component-Based Framework for the Formal Verification and Validation of Complex Aerospace Software

Pete Manolios, Northeastern University
Eric Feron, Georgia Tech
Cesar Munoz, National Institute of Aerospace

NASA Aviation Safety Technical Conference

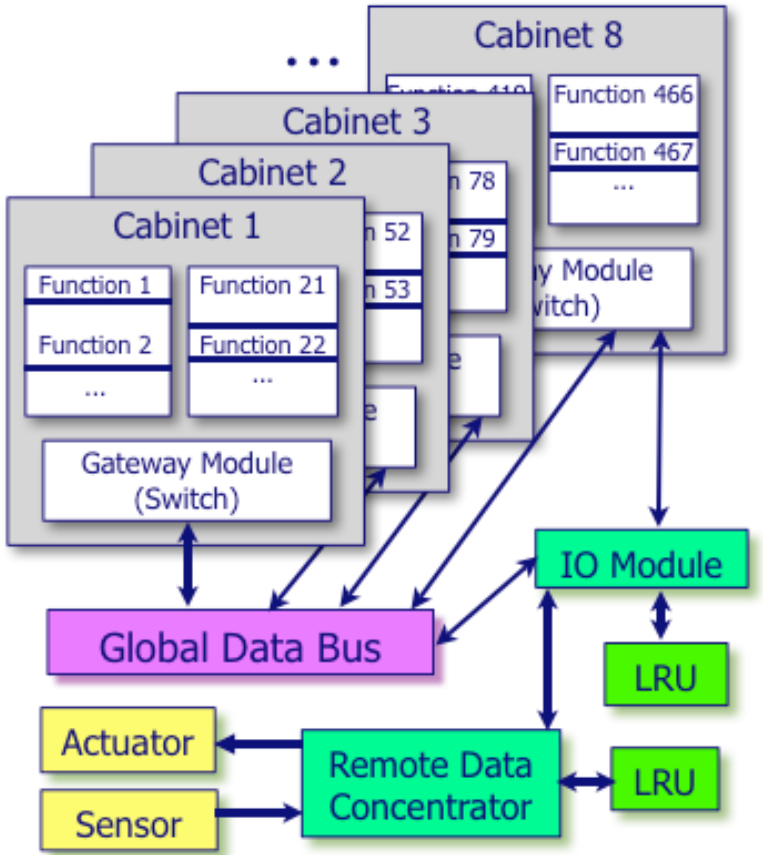
October 22, 2008

Project Overview

- Avionic systems comprised of many software components
- Important to ensure that
  - Individual components satisfy requirements
  - Connected, integrated & assembled to satisfy global requirements
- Formidable undertaking
  - Costly, error prone
  - Requires significant expert effort
  - Problem is getting worse as systems become more complex
- Develop a hierarchical component-based framework
  - Avionic systems are just too complex to verify monolithically

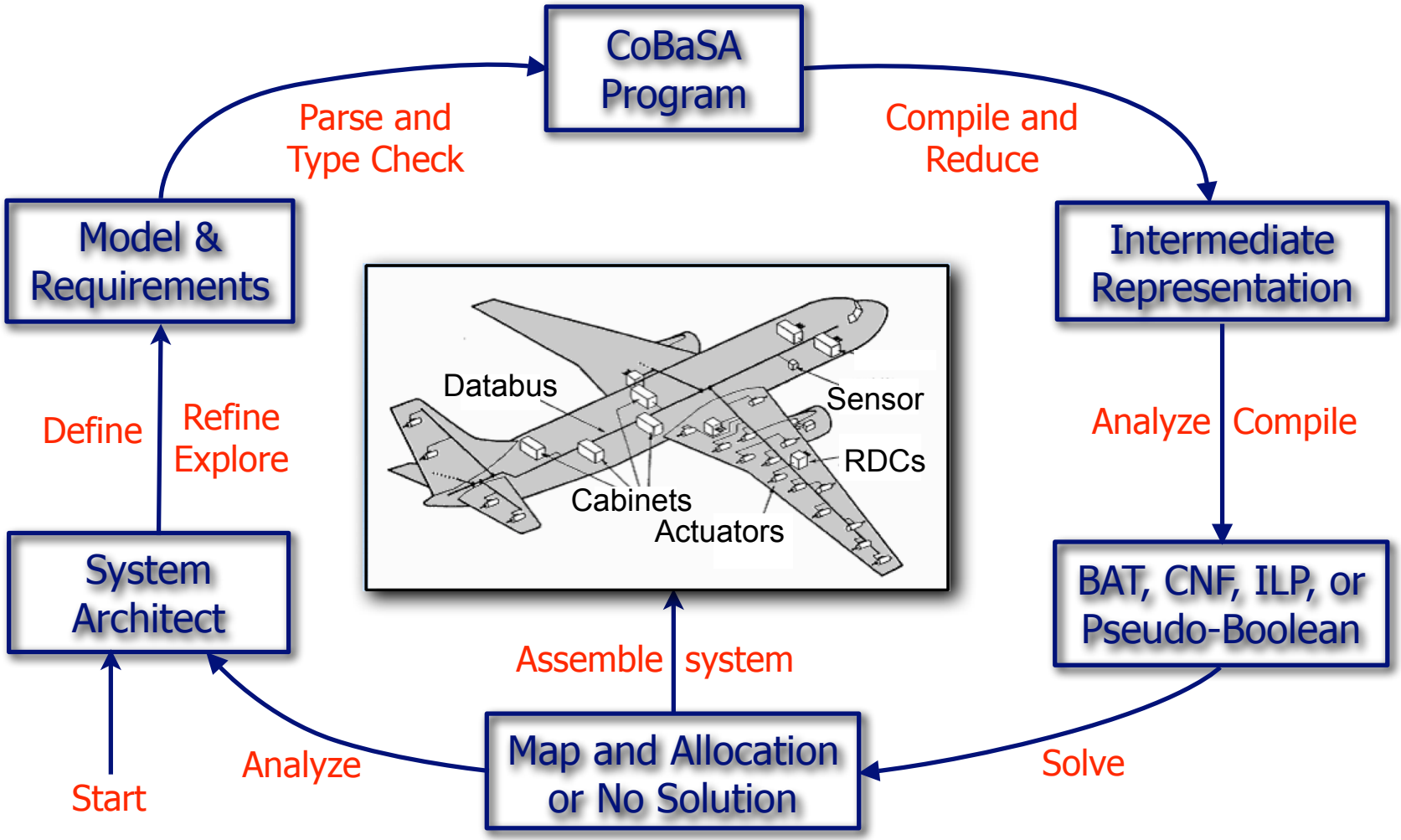
Component-Based System Assembly

- How do we ensure that the thousands of avionics components comprising the design of an airplane are:
- Connected, integrated, and assembled in a way that certifiably satisfies global requirements?
- Important problem
- Boeing's new business model
- Design and manufacturing of systems is shared with a global set of risk-sharing partners
- Boeing core competency (out of 3)

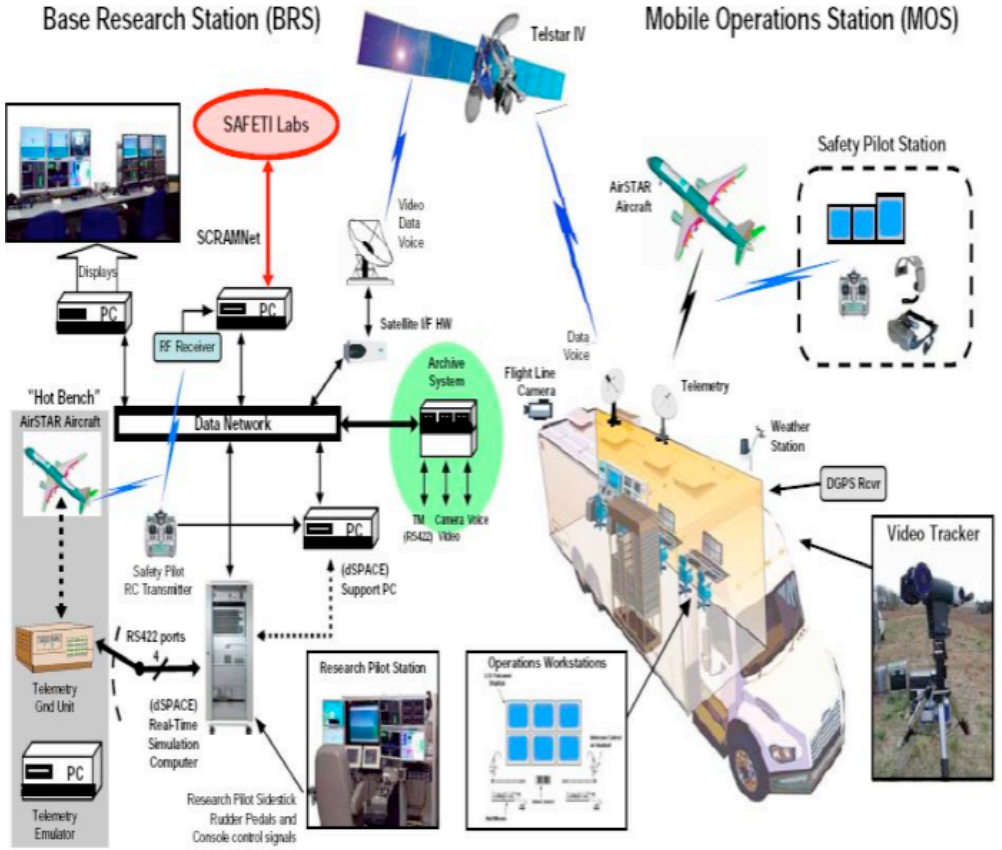


- Large-scale systems integration: We will continuously develop, advance, and protect the technical excellence that allows us to integrate effectively the systems we design and produce

Assembly of Avionics Systems



Hierarchical Verification of Distributed Protocols for Remotely-Operated Vehicles



AirSTAR is a dynamically scaled, remotely-operated vehicle developed at NASA LaRC to validate technologies under conditions that cannot be flight validated with full-scale vehicles.

AirSTAR Communication Protocol

- Flight commands are sent from pilots on the ground to the airborne vehicle
- Telementry from ROV is used by pilots to fly the vehicle and by research engineers for analysis and mission planning

Project Goals

- Design the AirSTAR communication protocol
- Two different protocols running in parallel:
  - Weak Delivery Protocol (WDP)
  - Guaranteed Delivery Protocol (GDP)
- Prove that it is correct with respect to safety and operational requirements
- Develop prototype
- Research Plan
  - Hierarchical verification
  - Develop tools for automating process
  - Automatic code generation

Control Systems Software Assurance

- We've got proofs for our control systems; let's use them!
- Goal: Certification of control algorithms, including adaptive algorithms
- Hypothesis: It is possible to convert stability and performance proofs of control system specifications into stability and performance proofs of control system codes, whether developed by hand or autocoded
- Plan: Leverage advanced performance/stability theories available for control system specifications to obtain code-level performance guarantees
- Formal link established between Lyapunov stability theory for both linear and nonlinear systems and Hoare's invariant theory

Relevance to IVHM

- Aviation Safety Program: Integrated Vehicle Health Management. Technical Plan, Version 2.01, August 2008
- NRC Decadal Survey of Civil Aeronautics R&T Challenges
  - Demonstrate compositional verification methods ... which ensures that individual software components individually satisfy their safety requirements, and, when connected, integrated, and assembled, satisfy global safety properties
  - Develop new methods and techniques for formal analysis of ultra-reliable distributed protocols and demonstrate the the new capability can accurately identify the classes and combinations of failures under which the architecture provides the correct services ...”
  - Provable correct protocols for fault-tolerant aviation communications systems

Observartions

- What technological barriers must be overcome to make SW Health Management feasible?
  - Automatic system assembly: high-level design and exploration
  - Adaptive assembly: assemble & reconfigure in real time
    - In response to system failure
    - Account environmental factors
    - Changes in mission priorities
    - Response to invalid assumptions?
    - Under extreme conditions (low power, long latencies, …)
- How do we prevent SW Health Management capabilities from becoming a source of problems? How do we guard against the guardian?

Observartions

- How do we prevent SW Health Management capabilities from becoming a source of problems? How do we guard against the guardian?
  - Example: in system assembly: garbage in is garbage out
  - As automation increases, this becomes a big risk
  - End-to-end arguments showing, with evidence, that overall, we have a safer system
- It is easy to make these arguments in a vacuum, we need validation from industry
  - Industry participation is necessary
  - We need creative ways to engage them